

Unit Interval Vertex Deletion: Fewer Vertices are Relevant*

Yuping Ke^{†‡}Yixin Cao[‡]Xiating Ouyang[‡]Jianxin Wang[†]

July 6, 2016

Abstract

The unit interval vertex deletion problem asks for a set of at most k vertices whose deletion from an n -vertex graph makes it a unit interval graph. We develop an $O(k^4)$ -vertex kernel for the problem, significantly improving the $O(k^{53})$ -vertex kernel of Fomin, Saurabh, and Villanger [ESA'12; SIAM J. Discrete Math 27(2013)]. We introduce a novel way of organizing cliques of a unit interval graph. Our constructive proof for the correctness of our algorithm, using interval models, greatly simplifies the destructive proofs, based on forbidden induced subgraphs, for similar problems in literature.

1 Introduction

A graph is a *unit interval graph* if its vertices can be assigned to unit-length intervals on the real line such that there is an edge between two vertices if and only if their corresponding intervals intersect. Given a graph G and an integer k , the *unit interval vertex deletion* problem asks whether there is a set of at most k vertices whose deletion makes G a unit interval graph. According to Lewis and Yannakakis [16], this problem is NP-complete.

This paper approaches this problem by studying its kernelization. Given an instance (G, k) of unit interval vertex deletion, a *kernelization algorithm* produces in polynomial time an “equivalent” instance (G', k') such that $k' \leq k$ and the *kernel size* (i.e., the number of vertices in G') is upper bounded by some function of k' . Fomin et al. [11] presented an $O(k^{53})$ -vertex kernel for the problem, which we improve to the following, where n and m denote respectively the numbers of vertices and edges of the input graph.

Theorem 1.1. *The unit interval vertex deletion problem has an $O(k^4)$ -vertex kernel, and it can be produced in $O(nm + n^2)$ time.*

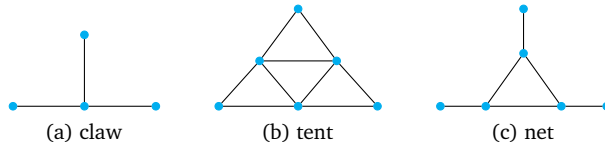


Figure 1: Forbidden induced graphs.

The structures of unit interval graphs have been well studied and well understood. It is known that a graph is a unit interval graph if and only if it contains no claw, net, tent, (as depicted in Figure 1,) or any hole (i.e., an induced cycle on at least four vertices) [19, 20]. One can decide in linear time whether a given graph is a unit interval graph; if it is not, we can obtain a forbidden induced subgraph in the same time [12]. The unit interval vertex deletion problem can then be equivalently defined as finding a set of at most k vertices that hits all forbidden induced subgraphs of the input graph.

*Supported in part by the Hong Kong Research Grants Council (RGC) under grant PolyU 252026/15E and the National Natural Science Foundation of China (NSFC) under grants 61572414 and 61420106009.

[†]School of Information Science and Engineering, Central South University, Changsha, China. jxwang@mail.csu.edu.cn.

[‡]Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. yixin.cao@polyu.edu.hk.

The vertex deletion problem has been defined on many other graph classes and has been intensively studied. It is the easy case when the objective graph class has a finite set of forbidden induced subgraphs: The sunflower lemma implies a polynomial kernel for the vertex deletion problem to this graph class [10]. However, kernels produced by the sunflower lemma tend to be very large. Furthermore, most interesting graph classes have an infinite number of forbidden induced subgraphs, hence the hard case. Another approach that works for both (and even edge modification problems) is to start from a modulator, i.e., a set of vertices whose deletion leaves a graph in the objective graph class. With a modulator, we are allowed to use the properties of unit interval graphs to analyze the rest of the graph. What is important is the interaction between other vertices with the modulator, and thus through its analysis we can identify irrelevant vertices, thereby producing the kernel [9, 15].

Since holes of any length are forbidden in unit interval graphs, our problem is clearly the hard case. Hence both Fomin et al. [11] and this paper use the modulator approach. For both of us, the modulator consists of two parts, first a set of vertices that hits all *small* forbidden induced subgraphs,—we use different thresholds for being small,—and the second an optimal hitting set for long holes in the remaining graph. Recall that long holes behave very nicely in a graph free of small forbidden induced subgraphs; for example, a minimum hitting set for them can be found in linear time [3]. Thus, our main concern is the first part. What differentiates these two algorithms is how they are carried out. Fomin et al. [11] used the sunflower lemma to produce the modulator, while we use a constant-approximation algorithm.

We only need to proceed when the approximation algorithm produces a solution of $O(k)$ vertices. Thus, our modulator has a linear size, which is in a sharp contrast with the huge modulator of Fomin et al. [11] produced by the sunflower lemma. On the other hand, their modulator has an extra property that is not shared by ours. It guarantees that one only needs to care about small forbidden induced subgraphs *inside the modulator*, thereby saving them a lot of trouble in the selection of relevant vertices. Our modulator nevertheless does not have this property. Therefore, the interaction of the modulator with the rest of the graph is far more complicated in our case, and the analysis is fundamentally different. In particular, the main technical difficulties present themselves exactly at the analysis of the small forbidden induced subgraphs.

This is exactly where our main technical ideas appear, which result in an algorithm and analysis significantly simpler than that of Fomin et al. [11]. Let (G, k) be the input instance and let M be the modulator. Our first idea is to *partition* the vertices of the unit interval subgraph $G - M$ into cliques and organize them in a linear way such that vertices in each clique are adjacent to only vertices in its two neighboring cliques. This is quite different from the widely used clique path decomposition, because in a clique path decomposition, (1) a vertex can appear in more than one cliques; and (2) two vertices in cliques that are far away can be adjacent. Our reduction rules ensure that if (G, k) is reduced, then there cannot be more than $O(k^2)$ cliques in the partition of $G - M$. From each of these cliques at most $O(k^3)$ vertices are relevant. This implies a kernel of $O(k^5)$ vertices, and a more careful analysis leads to the smaller size claimed in Theorem 1.1.

Our second idea appears in the proof of the correctness of our algorithm. Our reduction rules are rather elementary and self-explanatory. The main step is to prove the irrelevance of the other vertices. We may assume that (G, k) is already reduced, and let G' denote the subgraph induced by the relevant vertices and the modulator. We need to show that if there is a solution V_- of size at most k to our kernel G' , then it is a solution to G . Instead of showing the nonexistence of forbidden induced subgraph in $G - V_-$, (which would necessarily lead to an endless list of cases,) we build a unit interval model for $G - V_-$ out of a unit interval model for $G' - V_-$.

In a companion paper [5], we have also developed a polynomial kernel for the interval vertex deletion problem, which is arguably more challenging and was only recently shown to be fixed-parameter tractable [6]. It extends the ideas from this paper. It also starts from a modulator produced by the constant-approximation algorithm [4], but the analysis is far more complicated. In particular, the novel clique partition, which plays a crucial role in simplifying the analysis, does not apply there in any way we know of. Jansen and Pilipczuk [15] recently studied the kernelization of the chordal vertex deletion problem and produced the first polynomial kernel. They also used an approximation solution as the modulator, for which they had to first design a polynomial-time approximation algorithm. Their kernel has also a huge size, $O(k^{162})$ vertices.

Let us also mention the related parameterized algorithms (i.e., algorithms running in time $O(f(k) \cdot n^{O(1)})$ for some function f independent of n) for the problem, which have undergone a sequence of improvements. Recall that chordal graphs are those graphs containing no holes, and thus unit interval graphs are a subclass of chordal graphs. Toward a parameterized algorithm with $f(k) = \Omega(6^k)$, one can always dispose of all the claws, nets, and tents from the input graph, and then call the algorithm for the chordal vertex deletion problem [18, 7] to break all holes in the remaining graph. Direct algorithms for unit interval vertex deletion were later reported

in [1, 13, 3], and the current best algorithm runs in time $O(6^k \cdot (n + m))$. All the three direct algorithms use a two-phase approach. In [3], for example, the first phase breaks all claws, nets, tents, and C_4 's, while the second phase deals with the remaining $\{\text{claw, net, tent, } C_4\}$ -free graphs, on which the problem can be solved in polynomial time. A simple adaptation of this approach leads to an $O(nm + n^2)$ -time 6-approximation algorithm.

Organization. The rest of the paper is organized as follows. Section 2 introduces the clique partition and its properties. Sections 3 presents three simple reduction rules. Section 4 finishes the kernel by handpicking vertices from the reduced graph. Section 5 closes this paper by discussing implementation issues.

2 The clique partition

All graphs discussed in this paper are undirected and simple. A graph G is given by its vertex set $V(G)$ and edge set $E(G)$, whose cardinalities will be denoted by n and m respectively. All input graphs in this paper are assumed to be connected, hence $n = O(m)$ whenever $n > 1$. For $\ell \geq 4$, we use C_ℓ to denote a hole on ℓ vertices. Chordal graphs are precisely $\{C_\ell : \ell \geq 4\}$ -free graphs.

In this paper, all intervals are closed. An *interval graph* is the intersection graph of a set of intervals on the real line. The set of intervals, called an *interval model*, can be specified by their $2n$ endpoints: The interval $I(v)$ for vertex v is given by $[lp(v), rp(v)]$, where $lp(v)$ and $rp(v)$ are its *left* and *right endpoints* respectively. It always holds $lp(v) < rp(v)$. No distinct intervals are allowed to share an endpoint in the same model; note that this restriction does not sacrifice any generality. A graph is a *unit interval graph* if it has a *unit interval model*, where every interval has length one. An interval model is *proper* if no interval in it properly contains another interval. It is easy to see that every unit interval model is proper; a nontrivial observation of Roberts [19] is that every graph having a proper interval model also has a unit interval model.

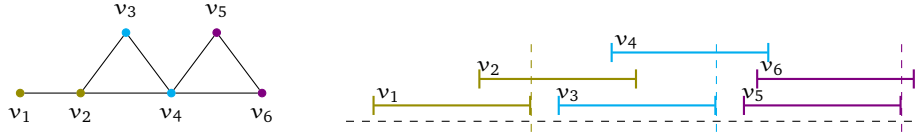


Figure 2: A unit interval graph and its unit interval model. The proper interval ordering decided by this model is $\langle v_1, v_2, v_3, v_4, v_5, v_6 \rangle$, which partitions the graph into cliques $\{v_1, v_2\}$, $\{v_3, v_4\}$, and $\{v_5, v_6\}$, corresponding to the three dashed vertical lines respectively.

Note that in a proper interval model, if $lp(u) < lp(v)$, then $rp(u) < rp(v)$ as well. Therefore, it makes sense to talk about the left-right relationship of the intervals. If we read the vertices by the ordering of their intervals, we end with a *proper interval ordering* of the graph [17]. The following property is an easy consequence of the definition of proper interval models and proper interval ordering.

Proposition 2.1. *Let v_1, \dots, v_n be a proper interval ordering of a unit interval graph G . For every $1 \leq i < j \leq n$, if $v_i v_j \in E(G)$, then $\{v_i, \dots, v_j\}$ is a clique.*

Fixing a unit interval model \mathcal{I} for a unit interval graph G , we can greedily partition its vertices into a sequence of cliques. Initially all vertices are unassigned. We repetitively choose the unassigned vertex v with the leftmost interval, and take all vertices whose intervals containing $rp(v)$; this set is clearly a clique. We proceed until the graph becomes empty. Let $\mathcal{K} = \{K_1, \dots, K_t\}$ be the set of cliques obtained in the order. See Figure 2 for an example. One should be noted that the cliques are not maximal in general; in particular the last vertex of K_{i-1} might be adjacent to all vertices in K_i , e.g., both the second the third cliques in Figure 2. The following proposition and its corollary characterize this partition, and facilitate our analysis of the kernel size.

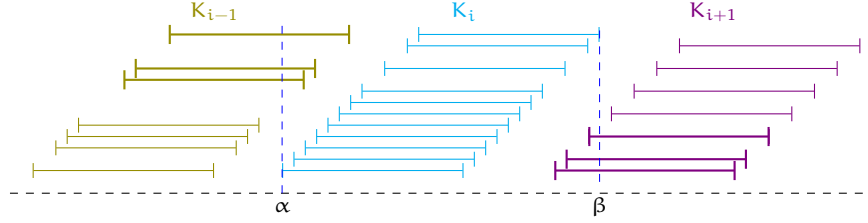
Proposition 2.2. *Let K_1, \dots, K_t be the clique partition of a unit interval model G . For each $1 < i < t$, it holds that $N(K_i) \subset K_{i-1} \cup K_{i+1}$. Moreover, $N(K_1) \subseteq K_2$, $N(K_t) \subset K_{t-1}$.*

Proof. Let $v \in K_i$. By the definition of clique partition, v is nonadjacent to the first vertex of K_{i-1} . By Proposition 2.1, no neighbor of v comes before the first vertex of K_{i-1} . Thus, if a neighbor of v is before K_i , it

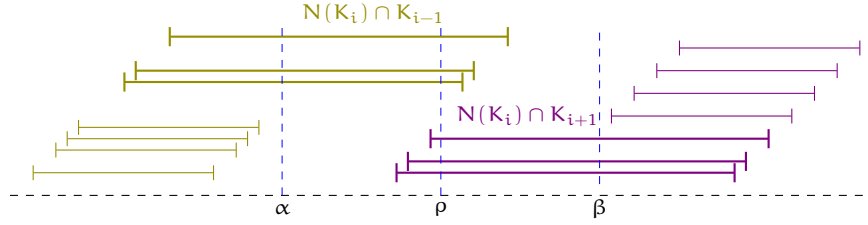
has to be in K_{i-1} . On the other hand, any neighbor of v after K_i is either the first vertex of K_{i+1} or adjacent to it, hence in K_{i+1} . The two border cases follow similarly. \square

Corollary 2.3. *Let K_1, \dots, K_t be the clique partition of a unit interval model G . For each pair of vertices $u \in K_i$ and $v \in K_j$ with $1 \leq i \leq j \leq t$, the distance between u and v is at least $j - i$.*

The following fact will be used in the correctness proof of our main reduction rule. Here by *contracting* a clique K_i ($1 < i < t$), we mean the operations of deleting all vertices in K_i , and adding all possible edges to connect its neighbors in K_{i-1} and in K_{i+1} .



(a) The original unit interval model for G .



(b) The new proper interval model for the graph obtained by contracting K_i . Only intervals for vertices in $N(K_i) \cap K_{i-1}$ and $N(K_i) \cap K_{i+1}$, which are thick, are extended.

Figure 3: Illustration for Proposition 2.4.

Proposition 2.4. *Let K_1, \dots, K_t be the clique partition of a unit interval graph G . For each $1 < i < t$, the graph obtained by contracting K_i is still a unit interval graph.*

Proof. Let G' be the graph obtained by contracting K_i ; then $V(G') = V(G) \setminus K_i$ and $E(G') = E(G - K_i) \cup ((N(K_i) \cap K_{i-1}) \times (N(K_i) \cap K_{i+1}))$. We build a proper interval model for G' as follows. Let α be the left endpoint of the first vertex in K_i , and let β be the right endpoint of the last vertex in K_i ; note that $\bigcup_{v \in K_i} I(v) = [\alpha, \beta]$. We choose an arbitrary point ρ between α and β . For vertices in $N(K_i) \cap K_{i-1}$, we change their right endpoints to be between ρ and β while keeping their orders. Likewise, for vertices in $N(K_i) \cap K_{i+1}$, we change their left endpoints to be between α and ρ while keeping their orders. See Figure 3. Note that only intervals for $N(K_i) \cap K_{i-1}$ and $N(K_i) \cap K_{i+1}$ are extended, and all the extensions are made in $[\alpha, \beta]$, where is disjoint from all other intervals. It is then easy to verify that the new interval model is proper and represents G' . \square

In passing we should point out that neither the ordering nor the clique partition is unique in general.¹

3 The reduction rules

Let (G, k) be an instance of the unit interval vertex deletion problem. We start by calling the 6-approximation algorithm [3] to find an approximation solution M to G . If $|M| > 6k$, then we return a trivial no-instance. We may assume henceforth $|M| \leq 6k$, and fix a unit interval model for $G - M$. Let $\sigma = \langle v_1, v_2, \dots, v_{n-|M|} \rangle$ be the proper interval ordering and $\mathcal{K} = \{K_1, \dots, K_t\}$ the clique partition derived from this model.

¹Even so, one can say that it is *almost unique*, in the sense that there can be at most two partitions: true twins (vertices with the same closed neighborhood) always reside in the same clique, while on a true-twin-free graph, there is only one ordering up to full reversal [8, 14].

As an easy consequence of Proposition 2.2, any vertex in M that is adjacent to five or more cliques in \mathcal{K} is the center of some claw. This observation inspires the following two reduction rules, whose correctness is straightforward: If you do not delete the vertex v itself, then you have to delete at least $k + 1$ vertices to break all claws involving v .

Rule 1. *If there exists a vertex $v \in M$ that is adjacent to at least $k + 5$ cliques in \mathcal{K} , then delete v and decrease k by 1.*

Rule 2. *If there exist a vertex $v \in M$ and at least five cliques in \mathcal{K} such that each of these cliques contains at least $k + 1$ neighbors of v , then delete v and decrease k by 1.*

Note that the diameter of a claw, net, or tent is at most three. The following is immediate from Corollary 2.3 and the fact that any claw, net, or tent of G needs to intersect M .

Corollary 3.1. *If there is $3 \leq \ell \leq t - 2$ such that M is nonadjacent to K_i for $\ell - 2 \leq i \leq \ell + 2$, then no vertex in K_ℓ is contained in a claw, net, or tent.*

Therefore, if there exists a long sequence of cliques that are nonadjacent to M , then most vertices in the middle can only participate in holes. Such a hole, if it exists, necessarily visits all these cliques, and in particular, it must enter from one end clique and leave at the other end. Moreover, it visits every clique in between, and contains one or two vertices from each of them. If we delete vertices from these cliques (for the purpose of breaking these holes), then we would choose a minimum separator. This observation motivates the next reduction rule. It has been observed in a more general form in [7, Reduction 2, Section 6]; with the clique partition, we can simplify it to the following form. Recall that any minimal separator of a unit interval graph is a clique, which cannot intersect more than two cliques in \mathcal{K} .

Rule 3. *Let K_{i-3}, \dots, K_{i+3} be 7 consecutive cliques in \mathcal{K} that are nonadjacent to M . Let u be the last vertex in K_{i-2} and let v be the first vertex in K_{i+2} . We take a minimum u - v separator S in $G - M$, and let $\ell \in \{i - 1, i, i + 1\}$ be that K_ℓ is disjoint from S . Contract K_ℓ .*

Lemma 3.2. *Rule 3 is safe: (G, k) is a yes-instance if and only if (G', k) is a yes-instance, where G' is the resulting graph.*

Proof. By Proposition 2.4, $G' - M$ is still a unit interval graph. Thus, every forbidden induced subgraph in G' needs to intersect M . Since we have only added edges between $K_{\ell-1}$ and $K_{\ell+1}$, for every vertex in them, its distance to M is at least three. There cannot be any claw, net, or tent in G' involving vertices from both $K_{\ell-1}$ and $K_{\ell+1}$. Therefore, we only need to take care of holes in the proof.

Suppose to the contrary of the only if direction that (G, k) is a yes-instance but (G', k) is not. Let V_- be a solution to (G, k) . Then there is necessarily a hole of G' that visits at least one edge added by the reduction; let it be xy with $x \in K_{\ell-1}$ and $y \in K_{\ell+1}$. Since x, y are nonadjacent to M , their other neighbors on the hole must both belong to $V(G) \setminus M$ as well; denote them by x' and y' respectively. The ordering of these four vertices has to be $x' <_\sigma x <_\sigma y <_\sigma y'$. There must be an x - y path in $G - M$ using only vertices in $K_{\ell-1}, K_\ell, K_{\ell+1}$. Its inner vertices are not adjacent to any vertex in this hole, except x, y themselves. Thus, we end with a hole of $G - V_-$, a contradiction.

On the other hand, if V_- is a solution to G' but there is hole in $G - V_-$, then there must be a hole visiting a vertex deleted by the reduction. This hole necessarily visit $N(K_\ell) \cap K_{\ell-1}$ and $N(K_\ell) \cap K_{\ell+1}$. But then after the reduction, its remaining vertices form a hole of $G' - V_-$: Note that the original hole has to visit M and hence has length larger than 4. \square

Lemma 3.3. *Each of the three reduction rules can be applied in $O(m)$ time.*

Proof. We can mark first the vertices in M , and then go through the adjacency list of each vertex in $G - M$ in the proper interval order. During this process we can record (1) for each vertex $v \in M$, how many cliques in \mathcal{K} are adjacent to v , and how many of them contain $k + 1$ or more neighbors of v ; and (2) for each clique $K \in \mathcal{K}$, whether it is adjacent to M . The process checks the adjacency list of each vertex once, and thus it takes $O(m)$ time in total. With this information, we can decide which of the three reduction rules is applicable, and if yes, apply it in the same time. \square

After the application of the reduction rules, it is possible that the rest of M is no longer a 6-approximation of the reduced graph. Therefore, we need to re-calculate the modulator. This would nevertheless take $O(n^2m)$ time. We defer the detailed for an efficient implementation to Section 5.

4 The kernel

Let (G, k) be a *reduced instance* with respect to modulator M , i.e., none of Rules 1–3 can be applied to G . Recall that σ is the fixed proper interval ordering of $G - M$, and $\mathcal{K} = K_1, \dots, K_t$ is the clique partition of $G - M$. We now pick up vertices from G to make the kernel. The idea is to pick as few as possible vertices that are relevant, i.e., from *each type of vertices* (to be defined later) we choose $k + 1$, which ensures that if any vertex from this type is not picked, then at least one picked vertex is not deleted by a solution of size at most k . Note that it is possible that there are less than $k + 1$ vertices in some type, and then we pick all of them. See Figure 4 for an example.

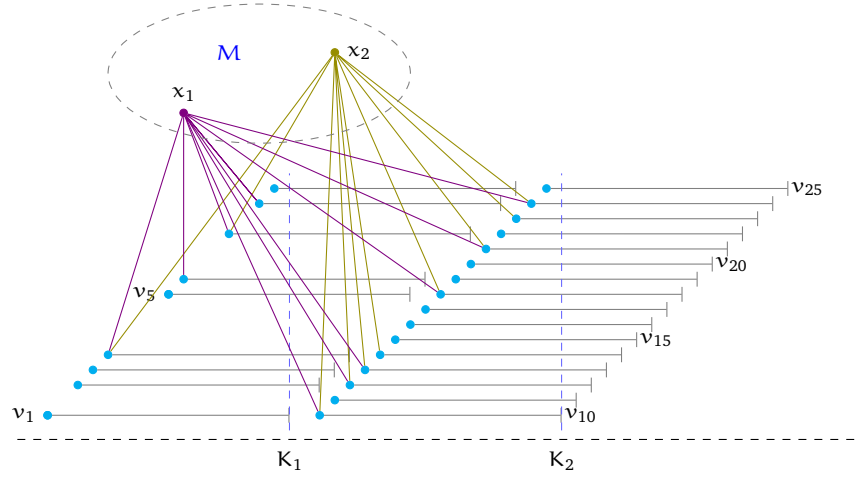


Figure 4: Illustration for picking vertices. The modulator M consists of x_1 and x_2 , and other 25 vertices are in $V(G) \setminus M$. Edges in $G - M$ are not drawn: The intervals, whose left endpoints coincide the vertices they represent, make a unit interval model for $G - M$. The vertices in $V(G) \setminus M$ are thus partitioned into two cliques, namely, $K_1 = \{v_1, \dots, v_9\}$ and $K_2 = \{v_{10}, \dots, v_{25}\}$. For $k = 2$, we have

$$K_2^1(x_1, x_2) = \{v_{10}, v_{12}, v_{13}\} \cup \{v_{18}, v_{21}, v_{24}\}; K_2^1(x_1, \bar{x}_2) = \emptyset; K_2^1(\bar{x}_1, x_2) = \{v_{14}, v_{23}\}; K_2^1(\bar{x}_1, \bar{x}_2) = \{v_{11}, v_{15}, v_{16}\} \cup \{v_{20}, v_{22}, v_{25}\}.$$

$$K_2^2(x_2, v_6) = \{v_{12}, v_{13}, v_{14}\}; K_2^2(x_2, v_8) = \{v_{14}, v_{18}, v_{21}\}; K_2^2(x_2, v_9) = \{v_{18}, v_{21}, v_{23}\}; K_1^2(x_2, v_{11}) = \{v_3, v_7\}; K_1^2(x_2, v_{15}) = K_1^2(x_2, v_{16}) = \{v_7\}.$$

$$K_3^3(x_1, v_4) = \{v_{13}, v_{18}, v_{21}\}; K_3^3(x_1, v_6) = \{v_{18}, v_{21}, v_{24}\}; K_3^3(x_1, v_7) = \{v_{21}, v_{24}\}; K_3^3(x_1, v_{24}) = \{v_6, v_7, v_8\}; K_1^3(x_1, v_{21}) = \{v_4, v_6, v_7\}; K_1^3(x_1, v_{18}) = \{v_4, v_6\}.$$

$$K_2^4(x_1, v_6) = \{v_{15}, v_{16}, v_{17}\}; K_2^4(x_1, v_7) = \{v_{17}, v_{19}, v_{20}\}; K_2^4(x_1, v_8) = \{v_{19}, v_{20}, v_{22}\}.$$

First, for each pair of vertices x_1, x_2 in M and each $i = 1, \dots, t$, we consider the (non)neighbors of x_1, x_2 in K_i . We pick the first and last $k + 1$ vertices from K_i for each of the four patterns—adjacent to both; adjacent to only x_1 ; adjacent to only x_2 ; and adjacent to neither. Let them be denoted by $K_i^1(x_1, x_2)$, $K_i^1(x_1, \bar{x}_2)$, $K_i^1(\bar{x}_1, x_2)$, and $K_i^1(\bar{x}_1, \bar{x}_2)$ respectively. Also, let $K_i^1(x)$ denote the first $k + 1$ and the last $k + 1$ of $\bigcup_{y \in M \setminus \{x\}} (K_i^1(x, y) \cup K_i^1(x, \bar{y}))$.

Second, For each $x \in M$, each $i = 2, \dots, t$, and each of the last $k + 1$ non-neighbors y of x in K_{i-1} , we pick the last $k + 1$ common neighbors of x and y in K_i ; for each $x \in M$, each $i = 1, \dots, t - 1$, and each of the first $k + 1$ non-neighbors y of x in K_{i+1} , we pick the first $k + 1$ common neighbors of x and y in K_i . Let them be denoted by $K_i^2(x, y)$.

Third, for each $x \in M$, each $i = 2, \dots, t$, and each of the first $k + 1$ neighbors y of x in K_{i-1} , we pick the first $k + 1$ vertices in K_i that are neighbors of x but not y ; for each $x \in M$, each $i = 1, \dots, t - 1$, and each of the last $k + 1$ neighbors y of x in K_{i+1} , we pick the last $k + 1$ vertices in K_i that are neighbors of x but not y . Let them be denoted by $K_i^3(x, y)$.

Fourth, for each $x \in M$, each $i = 2, \dots, t$, and each of the last $k + 1$ neighbors y of x in K_{i-1} , we pick the last $k + 1$ vertices in K_i that are neighbors of y but not x ; for each $x \in M$, each $i = 1, \dots, t - 1$, and each of the first

$k + 1$ neighbors y of x in K_{i+1} , we pick the first $k + 1$ vertices in K_i that are neighbors of y but not x . Let them be denoted by $K_i^4(x, y)$.

Finally, for each three pairwise nonadjacent vertices in M , we arbitrarily pick $k + 1$ common neighbors of them in $V(G) \setminus M$; and for each triple of vertices in M that induces a P_3 , we arbitrarily pick $k + 1$ vertices in $V(G) \setminus M$ that are adjacent to only the center vertex among them. Let them be denoted by V_0 .

Let K be a clique in \mathcal{K} . If $|K| \leq 2k + 2$, then all its vertices have been picked. We consider then the nontrivial case, i.e., when $|K| > 2k + 2$. The first and last $k + 1$ vertices of K are always picked; hence at least $2k + 2$ vertices are picked from K . Likewise, the first and the last $k + 1$ vertices in K that are nonadjacent to M are always picked; so are the first and the last $k + 1$ neighbors in K for each $x \in M$. Moreover, if a vertex v satisfies the conditions of any particular set but is not picked, then we have picked from the set $2(k + 1)$ vertices, of which $k + 1$ are to the left of v , and $k + 1$ are to the right of v .

Let G' be induced by the picked vertices together with M . We now calculate the cardinality of $V(G')$. There are $O(\binom{|M|}{2} * O(k) + O(|M|) * O(k) * O(k) = O(k^3))$ vertices picked from each clique. On the other hand, the number t of cliques in \mathcal{K} is $O(k^2)$, as otherwise one of Rules 1 and 3 must be applicable. Together with at most $(\binom{|M|}{3} * (k + 1) * 2 = O(k^4))$ vertices in V_0 , and $O(k)$ vertices in M , a rough estimation of $|V(G')|$ would be $O(k^5)$. A refined analysis would bring it to $O(k^4)$.

Lemma 4.1. *The new graph G' has at most $O(k^4)$ vertices.*

Proof. Since Rule 1 is not applicable, for each $v \in M$ there are at most $k + 5$ cliques intersecting $N(v)$. There are at most $|M| \times (k + 5) = O(k^2)$ cliques adjacent to M . On the other hand, since Rule 3 is not applicable, at most 6 consecutive cliques can be nonadjacent to M . Therefore, the number t of cliques in \mathcal{K} is at most $O(k^2)$.

We consider first the vertices that are not in $N[M]$. In the first category, we choose from each clique at most $2k + 2$ vertices that are nonadjacent to M . In the third and fourth categories, we choose from each clique at most $|M| \times (k + 1) * 4 = O(k^2)$ vertices that are nonadjacent to M . Therefore, $|V(G') \setminus N[M]| = O(k^2) * O(k^2) = O(k^4)$.

Consider then vertices in $N(M)$. Since Rule 2 is not applicable, for each $v \in M$ there can be at most four cliques containing $k + 1$ or more vertices from $N(v)$. There are thus at most $|M| \times 4 \leq 24k$ such cliques. From each of them we picked $O(\binom{|M|}{2} * O(k) + O(|M|) * O(k) * O(k) = O(k^3))$ vertices, and hence the total number of vertices picked from these cliques is $O(k^4)$. Each of the other cliques contains at most k neighbor of each vertex $v \in M$, and no more than $|M| \times k = O(k^2)$ vertices from $N(M)$. Therefore from these cliques we picked at most $O(k^2) \times O(k^2) = O(k^4)$ vertices that are neighbors of M .

In summary,

$$|V(G')| = |V(G') \setminus N[M]| + |V(G') \cap N(M)| + |M| = O(k^4) + O(k^4) + O(k) = O(k^4).$$

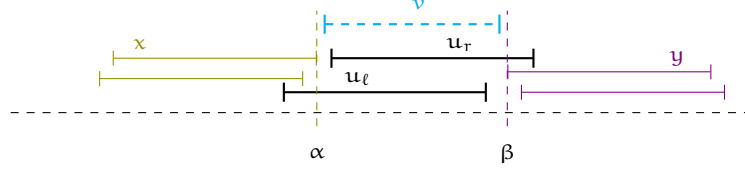
The proof is now complete. \square

To conclude Theorem 1.1, it remains to verify the equivalence between the new instance (G', k) and the original instance. Similar as the proof of Proposition 2.4, the proof of our main lemma would be manipulating intervals. We also take liberty to produce a proper interval model instead of a unit interval model: One can always turn it into a unit interval model by, say, calling the algorithm of Bogart and West [2]. Another trick we want to play is the following. Since the set of endpoints is finite, for any point ρ in an interval model, we can find a small positive value ϵ such that there is no endpoint in $[\rho - \epsilon, \rho) \cup (\rho, \rho + \epsilon]$,—in other words, there is an endpoint in $[\rho - \epsilon, \rho + \epsilon]$ if and only if ρ itself is an endpoint. Note that the value of ϵ should be understood as a function, depending on the interval model as well as the point ρ , instead of a constant.

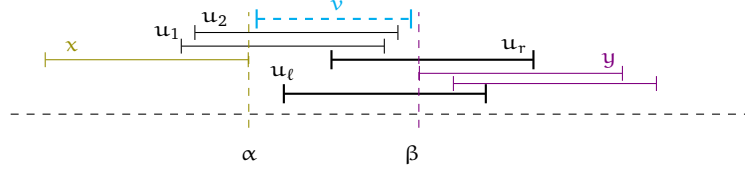
Lemma 4.2. *If there is V_- with $|V_-| \leq k$ such that $G' - V_-$ is a unit interval graph, then $G - V_-$ is also a unit interval graph.*

Proof. We build a proper interval model for $G - V_-$ by inserting intervals for $V(G) \setminus V(G')$ into a unit interval model for $G' - V_-$ as follows. Note that G' contains M , and hence all vertices in $V(G) \setminus V(G')$ appear in σ , which is a proper interval ordering for $G - M$. Let $G_0 = G'$, and let G_i , for $1 \leq i \leq |V(G) \setminus V(G')|$, denote the subgraph induced by $V(G')$ and the first i vertices of $V(G) \setminus V(G')$ in σ .² Since G_0 is a unit interval graph, by inductive reasoning, it suffices to show how to build a proper interval model for $G_i - V_-$ out of $G_{i-1} - V_-$.

²Our construction in the proof does not rely on any particular ordering, and it can be arbitrary.



(a) No vertex is adjacent to both u_ℓ and u_r but not v .



(b) Some non-neighbor of v is adjacent to both u_ℓ and u_r from the right.

Figure 5: Illustration for Lemma 4.2.

Let v be the i th vertex of $V(G) \setminus V(G')$, and let \mathcal{I} be a unit interval model for $G_{i-1} - V_-$. Let K_b be the clique in the clique partition of $G - M$ that contains v . Since v itself is not in G' , we have picked from K_b the first $k+1$ and the last $k+1$ vertices; from each of them at least one vertex is not in V_- . There are thus vertices u_ℓ, u_r in $G_{i-1} - (V_- \cup M)$ such that $u_\ell <_\sigma v <_\sigma u_r$. Assume without loss of generality $I(u_\ell)$ is to the left of $I(u_r)$.

Consider first that $N_{G_i}(u_\ell) \setminus N_{G_i}[v]$ and $N_{G_i}(u_r) \setminus N_{G_i}[v]$ are disjoint, i.e., for every vertex $u \in V(G_{i-1}) \setminus V_-$ that is nonadjacent to v , the interval $I(u)$ intersects at most one of $I(u_\ell)$ and $I(u_r)$. Let x be the vertex in $N_{G_i}(u_\ell) \setminus N_{G_i}[v]$ with the rightmost interval, and let y be the vertex in $N_{G_i}(u_r) \setminus N_{G_i}[v]$ with the leftmost interval; denote by $\alpha = \text{rp}(x)$ and $\beta = \text{lp}(y)$. See Figure 5(a). Then

$$\text{lp}(u_\ell) < \alpha < \text{lp}(u_r) < \text{rp}(u_\ell) < \beta < \text{rp}(u_r),$$

and every vertex in $G_{i-1} \setminus V_-$ with its interval properly contained in $[\alpha, \beta]$ has the same closed neighborhood as v in $G_i \setminus V_-$.

We now argue that no interval can contain $[\alpha, \beta]$. Suppose for contradiction $[\alpha, \beta] \subseteq I(u)$, then $\{u, x, y, v\}$ is a claw of G_i . At least one of these four vertices is in M , because $G - M$ is a unit interval graph. Noting that v is not in M , we consider which of u, x, y are in M . Note that the other vertices in $V(G_i) \setminus (V_- \cup M)$ may or may not be in G' .

- Case 1, $x, y, u \in M$. In V_0 there are at least $k+1$ vertices each of which makes a claw with $\{u, x, y\}$. At least one of them is not in V_- and hence $G' - V_-$ contains a claw.
- Case 2, $x, y \in M$ but $u \notin M$. Let $u \in K_a$. We may assume $u <_\sigma v$; then $a = b$ or $b - 1$: Since u and v are adjacent, they are either in the same clique or in two consecutive cliques in the partition \mathcal{K} . We take u' to be the last vertex of $K_a^1(x, y) \setminus V_-$ (it is nonempty because $|K_a^1(x, y)| > k$ when u is not in it), and take v' to be the first of $K_b^1(\bar{x}, \bar{y}) \setminus V_-$. We claim that $\{u', v', x, y\}$ is always a claw in $G' - V_-$. By the selection, it suffices to verify that $u'v' \in E(G)$. It is trivial when $a = b$, and it follows from $u \leq_\sigma u' <_\sigma v' <_\sigma v$ and Proposition 2.1 when $a = b - 1$.
- Case 3, u and one of x, y is in M . We consider x and the other is symmetric. Let $y \in K_c$; note that $c \neq b$ because y and v are nonadjacent. We may assume $b < c$. We take v' to be the first of $K_b^1(u, \bar{x}) \setminus V_-$, and take y' to be the last of $K_c^1(u, \bar{x}) \setminus V_-$. They are nonadjacent because $v' <_\sigma v <_\sigma y \leq_\sigma y'$ and $vy \notin E(G)$. Then $\{u, v', x, y'\}$ is a claw in $G' - V_-$.
- Case 4, only u is in M . Let $x \in K_a$ and $y \in K_c$; note that a, b , and c are all distinct because v, x , and y are pairwise nonadjacent. We may assume $a < b < c$ (i.e., $x <_\sigma v <_\sigma y$). We take x' to be the first of $K_a^1(u) \setminus V_-$, and take y' to be the last of $K_c^1(u) \setminus V_-$. They are clearly adjacent to u but nonadjacent to each other.

- 4.1. If $b > a + 1$, then we take v' to be the first of $K_b^1(u) \setminus V_-$; it is nonadjacent to x' . It is also nonadjacent to y' because $v' <_\sigma v <_\sigma y \leq_\sigma y'$ and $vy \notin E(G)$.
- 4.2. Otherwise, we take v' to be the first of $K_b^3(u, x') \setminus V_-$. Note that $x' \leq_\sigma x <_\sigma v$; by Proposition 2.1 x' is nonadjacent to v . As a result, $K_b^3(u, x') \setminus V_-$ is nonempty and $v' <_\sigma v$. Together with $v <_\sigma y \leq_\sigma y'$ and $vy \notin E(G)$, we have $v'y' \notin E(G)$. The definition of $K_b^3(u, x')$ implies $x'v' \notin E(G)$.

Therefore, $\{u, v', x', y'\}$ is always a claw in $G' - V_-$.

- Case 5, only one of x, y is in M . We consider x and the other is symmetric. Then vuy is a P_3 in $G_i - V_-$; we may assume $v <_\sigma u <_\sigma y$. Clearly, v and y are nonadjacent and hence in different cliques.
 - 5.1. If none of them is in the same clique as u , then $u \in K_{b+1}$ and $y \in K_{b+2}$. We take v' to be the last of $K_b^1(\bar{x}) \setminus V_-$; take u' to be the last of $K_{b+1}^2(x, v') \setminus V_-$; and take y' to be the first of $K_{b+2}^1(\bar{x}) \setminus V_-$. The vertex v' is clearly adjacent to u' but not y' . To see that v' is not adjacent to y' , note $u \leq_\sigma u' <_\sigma y' \leq_\sigma y$.
 - 5.2. Otherwise, $y \in K_{b+1}$ and u is in either K_b or K_{b+1} . Assume without loss of generality that $u \in K_{b+1}$. We take u' to be the first of $K_{b+1}^1(x) \setminus V_-$, take v' to be the first of $K_b^4(x, u') \setminus V_-$, and take y' to be the last of $K_{b+1}^1(\bar{x}) \setminus V_-$.

In either case, $\{u', v', x, y'\}$ is a claw of $G' - V_-$.

Therefore, if no interval of J is contained in $[\alpha, \beta]$, then making $I(v) = [\alpha + \epsilon, \beta - \epsilon]$ would make a proper interval model for G_i . Otherwise let $[\alpha', \beta']$ be such an interval contained in $[\alpha, \beta]$; we can make $I(v) = [\alpha' + \epsilon, \beta' + \epsilon]$.

In the rest there exists at least one non-neighbor u of v such that $I(u)$ intersects both $I(u_\ell)$ and $I(u_r)$. We argue that $I(u)$ cannot be contained in $[lp(u_\ell), rp(u_r)]$. Suppose such a vertex u exists, then it must be from M : because $u_\ell <_\sigma v <_\sigma u_r$, no vertex in the unit interval graph $G - M$ can be adjacent to both u_ℓ and u_r but not v . Since the model is proper, if $I(u) \subseteq [lp(u_\ell), rp(u_r)]$, then $[lp(u_r), rp(u_\ell)] \subseteq I(u)$. But we had also chosen from K the first $k + 1$ and the last $k + 1$ non-neighbors of u . At least one of these vertices remains in G_i and its interval has to be intersect both $I(u_\ell)$ and $I(u_r)$ but not $I(u)$. This is impossible.

Therefore, $I(u)$ approaches $I(u_\ell)$ and $I(u_r)$ either from the left or the right. We may assume without loss of generality it is to the right of $I(u_r)$ (i.e., $rp(u_r) \in I(u)$); the other case follows by symmetry. Let us take the non-neighbor y of v in G_i that has the leftmost interval containing $rp(u_r)$; let $\beta = lp(y)$. Then $\beta = \min_u \{lp(u) : u \in V(G_i) \setminus N(v), u_r \in I(u)\}$, and by assumption, $\beta \in I(u_\ell)$. Since $u_\ell <_\sigma v <_\sigma u_r$, the vertex y has to be from M . But then we would have also chosen from K the first $k + 1$ and the last $k + 1$ non-neighbors of y . At least one from either set is in G_i ; let them be u_1 and u_2 respectively. The intervals $I(u_1)$ and $I(u_2)$ have to approach $I(u_\ell)$ from the left. Again, there cannot be vertices from $G_i - M$ adjacent to both u_1, u_2 but not v . We argue that for any vertex $x \notin N(v)$ adjacent to u_1 and/or u_2 , the interval $I(x)$ is disjoint from and to the left of $I(u_\ell)$ (i.e., $rp(x) < lp(u_\ell)$). We have also chosen from K the first $k + 1$ and the last $k + 1$ vertices that are adjacent to neither x nor y . At least one of them is in $G_i - V_-$ and its interval has to be accommodated between $(rp(x), lp(y))$. It would then be properly contained in $I(u_\ell)$ if $rp(x) > lp(u_\ell)$. Let $\alpha = \max_u \{rp(u) : u \in V(G_i) \setminus N(v), lp(u_1) \in I(u)\}$. See Figure 5(b). The rest of the construction is the same as the first one. \square

5 Implementation issues and concluding remarks

In principle, each application of Rules 1–3 should be followed by a re-calculation of the modulator: After the application, the rest of M (it loses one vertex with Rules 1 and 2 but remains intact with Rules 3) may not be a 6-approximation for the remaining graph. This would imply that it takes $O(n \cdot nm)$ time to exhaustively apply Rules 1–3.

We have been using the approximation algorithm [3] as a black box for furnishing the modulator. To have a better analysis, we may have to unwrap the black box and see a bit of how it works. It consists of two phases. The first phase keeps looking for a claw, net, tent, C_4 , or C_5 , and deletes all its vertices if one is found. When none of these small forbidden induced subgraphs can be found, the algorithm enters the second phase, which then finds an optimal solution in linear time. The ratio is 6 because unit interval graphs are hereditary and any optimal solution needs to contain at least one vertex from any induced claw, net, tent, C_4 , or C_5 . Recall

that whether a graph contains a claw, net, tent, C_4 , or C_5 can be decided in linear time, and if yes, one can be detected in the same time.

Consider first Rules 1 and 2, each of which deletes a vertex from M . If the deleted vertex v had been added to M in the second phase of the approximation algorithm, then the set $M - \{v\}$ is still a 6-approximation for $G - \{v\}$, and we need to do nothing. Otherwise, we need to (re-)calculate a new approximation solution for $G - \{v\}$. Fortunately, we do not need to start from scratch. Recall that v had been put into M because it is in some induced claw, net, tent, C_4 , or C_5 found in phase 1; let X be the at most six vertices of this forbidden induced subgraph. Let M' denote the subset of vertices of $M \setminus X$ that are added in the first phase; they are still *good* in the sense that they still form vertex-disjoint claws, nets, tents, C_4 's, and C_5 's. Therefore, we may start the approximation algorithm with M' as the partial solution. Note that every claw, net, tent, C_4 , or C_5 in $G - \{v\} - M'$ needs to intersect $X \setminus \{v\}$. Therefore, we can find at most six vertex-disjoint claws, nets, tents, C_4 's, and C_5 's, which can be done in $O(m) * 6 = O(m)$ time. We put all the vertices in the found subgraphs, and redo the second phase in another $O(m)$ time. Consequently, we can produce a 6-approximation for the new graph $G - \{v\}$ in $O(m)$ time.

The situation for Rule 3 is actually simpler. It does not touch M , and thus all vertices added in the first phase remain good. We can redo the second phase in $O(m)$ time to produce an approximation solution for the new graph.³

Therefore, we can apply each reduction rule and presently recover the modulator in $O(m)$ time. On the other hand, since each application of a reduction rule deletes at least one vertex from the graph, they can be applied at most n times. The total running time of them is $O(nm)$. The picking of the vertices can be easily done in $O(k^3m)$ time. Note that if $n < k^4$, then we do not need to do anything at all. Therefore, the running time of the whole kernelization algorithm is $O(nm)$.

The primary concern of a kernelization algorithm is surely the kernel size. Kernelization algorithms may not completely solve the instance, and then they are followed by other algorithmic approaches. Its applicability would thus be limited if the running time is too high. In literature, however, very little attention has been paid to the running time of most kernelization algorithms, and most of the time, a detailed analysis is omitted. (Most of them are trivially polynomial.) Once we aim for “efficient” kernelization with lower polynomial running time, we need to reconsider the tools we can use. For example, for all vertex deletion problems to hereditary graph class, we have a trivially correct reduction rule that deletes all vertices not participating in any forbidden induced subgraphs. This, however, is usually very time-consuming (it takes normally $n^{|X|}$ where X is the largest forbidden induced subgraph) and thus should be avoided.

As a final remark, properties of the approximation algorithm [3] may be further exploited to sharpen the analysis of the kernel size of our kernelization algorithm. But it would very unlikely lead to one with $o(k^2)$ vertices. We leave the existence of a linear-vertex kernel for the unit interval vertex deletion problem as an open question.

Acknowledgments. The authors want to thank Jinshan Gu, R. B. Sandeep, and Jie You for fruitful discussions during an early stage of this project.

References

- [1] René van Bevern, Christian Komusiewicz, Hannes Moser, and Rolf Niedermeier. Measuring indifference: Unit interval vertex deletion. In Dimitrios M. Thilikos, editor, *Graph-Theoretic Concepts in Computer Science (WG)*, volume 6410 of *LNCS*, pages 232–243. Springer, 2010. doi:10.1007/978-3-642-16926-7_22.
- [2] Kenneth P. Bogart and Douglas B. West. A short proof that ‘proper = unit’. *Discrete Mathematics*, 201(1-3):21–23, 1999. doi:10.1016/S0012-365X(98)00310-0.
- [3] Yixin Cao. Unit interval editing is fixed-parameter tractable. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages and Programming (ICALP)*, volume 9134 of *LNCS*, pages 306–317. Springer, 2015. Full version available at arXiv:1504.04470. doi:10.1007/978-3-662-47672-7_25.

³This step is not really necessary, because it can be proved that M remains a 6-approximation of the new graph after the application of Rule 3. Moreover, the impact of Rule 3 on the interval model and the clique partition of $G - M$ is local, and a new model and a new partition can be easily recovered. For the simplicity of presentation, the form of Rule 3 given in Section 3 deletes only one clique. One can show that it can be easily adapted to contracting all but 6 cliques in a sequence of cliques in \mathcal{K} that are nonadjacent to M . Moreover, all the cliques can be handled in one run, in linear time. To prove these facts, however, we need to revisit the approximation algorithm [3] with all the details, which we omit to not blur the focus of the current paper.

- [4] Yixin Cao. Linear recognition of almost interval graphs. In Robert Krauthgamer, editor, *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1096–1115. SIAM, 2016. Full version available at arXiv:1403.1515. doi:10.1137/1.9781611974331.ch77.
- [5] Yixin Cao, Yuping Ke, and Xiating Ouyang. A polynomial kernel for interval vertex deletion. Manuscript, 2016.
- [6] Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Transactions on Algorithms*, 11(3):21:1–21:35, 2015. A preliminary version appeared in SODA 2014. doi:10.1145/2629595.
- [7] Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016. A preliminary version appeared in STACS 2014. doi:10.1007/s00453-015-0014-x.
- [8] Xiaotie Deng, Pavol Hell, and Jing Huang. Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. *SIAM Journal on Computing*, 25(2):390–403, 1996. doi:10.1137/S0097539792269095.
- [9] Pål Grønås Drange and Michal Pilipczuk. A polynomial kernel for trivially perfect editing. In Nikhil Bansal and Irene Finocchi, editors, *Algorithmics (ESA)*, volume 9294 of *LNCS*, pages 424–436. Springer, 2015. Full version available at arXiv:1412.7558. doi:10.1007/978-3-662-48350-3_36.
- [10] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- [11] Fedor V. Fomin, Saket Saurabh, and Yngve Villanger. A polynomial kernel for proper interval vertex deletion. *SIAM Journal on Discrete Mathematics*, 27(4):1964–1976, 2013. A preliminary version appeared in ESA 2012. doi:10.1137/12089051X.
- [12] Pavol Hell and Jing Huang. Certifying LexBFS recognition algorithms for proper interval graphs and proper interval bigraphs. *SIAM Journal on Discrete Mathematics*, 18(3):554–570, 2004. doi:10.1137/S0895480103430259.
- [13] Pim van ’t Hof and Yngve Villanger. Proper interval vertex deletion. *Algorithmica*, 65(4):845–867, 2013. doi:10.1007/s00453-012-9661-3.
- [14] Wen-Lian Hsu. $O(m \cdot n)$ algorithms for the recognition and isomorphism problems on circular-arc graphs. *SIAM Journal on Computing*, 24(3):411–439, 1995. doi:10.1137/S0097539793260726.
- [15] Bart M. P. Jansen and Marcin Pilipczuk. Approximation and kernelization for chordal vertex deletion. arXiv:1605.03001, 2016.
- [16] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. Preliminary versions independently presented in STOC 1978. doi:10.1016/0022-0000(80)90060-4.
- [17] Peter J. Looges and Stephan Olariu. Optimal greedy algorithms for indifference graphs. *Computers & Mathematics with Applications*, 25(7):15–25, 1993. doi:10.1016/0898-1221(93)90308-I.
- [18] Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010. A preliminary version appeared in WG 2006. doi:10.1007/s00453-008-9233-8.
- [19] Fred S. Roberts. Indifference graphs. In Frank Harary, editor, *Proof Techniques in Graph Theory (Proc. Second Ann Arbor Graph Theory Conf., 1968)*, pages 139–146. Academic Press, New York, 1969.
- [20] Gerd Wegner. *Eigenschaften der Nerven homologisch-einfacher Familien im R^n* . PhD thesis, Universität Göttingen, 1967.